REMARKS

Claims 20-39 remain in this application. Claims 20, 25, 29, 31, 34, and 36 have been amended. First, the Applicants would like to express their gratitude toward the Examiner for taking the time to discuss the present application by telephone (on October 15, 2003) prior to this submission. The amendments herein reflect the subject matter discussed by telephone with the Examiner and are believed to place the claims in condition for allowance.

In the telephone discussion, the Examiner agreed to withdraw the 35 U.S.C. §112 rejections regarding the "stateless DTU." However, the Examiner maintained her rejection to Claim 31 regarding the need to clarify the interconnectivity of the "filter" and the "resource" limitations. See para. 5, page 2 of the Office Action. Although the Applicants respectfully believe that the elements recited in Claim 31 are interconnected, the Applicants, nevertheless, rewrite Claim 31 to further clarify the subject matter being claimed and to expedite allowance. Specifically, Claim 31 has been rewritten to now read:

> **a resource;**
>
> **a filter for managing consumption of said resource;**
>
> wherein said filter is separated from said plurality of applications;
>
> * * *
>
> a first signal transmitted **from said filter to at least one member of said plurality of applications** indicating that **said at least one member** should stop **consuming said resource;**
>
> * * *
>
> a second signal transmitted from said filter to said at least one member indicating that said at least one member should resume consuming said resource . . . (Emphasis in underline added).

In view of the foregoing, the Applicants respectfully submit that the rejection under 35 U.S.C. §112, first paragraph, should be withdrawn. Moreover, as indicated below, Claim 31 should now be allowable.

Before responding to the Examiner's rejection based on the prior art, a brief description of the present application is provided. The present application is directed toward a method and apparatus for utilizing resources on a shared client network environment. Computers in a network environment can be categorized as two types: servers and clients. In addition, a client can be further understood to be a stateless thin client (in contrast with a thick client or a full-featured workstation). A stateless thin client (or a stateless DTU) is a small, stateless, "plug and work" desktop computer whose main function is to process all input and output for the user and to manage communication with at least one server. All other client processing for the user are concentrated on a group of client servers and shared amongst a community of DTUs. The group of client servers can be called a shared client (or a consolidated client) because, although the servers are often the equivalent of larger powerful server machines, they perform the traditional role of the traditional "client" in a traditional client/server architecture. In addition, the shared client is "shared" by a large number of DTUs (that are shared by an even larger number of users on the DTUs). The removal of the traditional client processing (e.g., state maintenance and computation power) from the DTU (or thin client) into the shared client servers permits simplification of the DTU in the network because software and hardware for performing these tasks are not needed at the DTU.

Because the DTUs of the present invention are stateless (i.e., devices that process information without any knowledge of previous/subsequent information), a user's interaction with the network are managed using a persistent user session and the interaction can be instantly sent to any DTU within the network. That is, a user can be in the middle of a user session (associated with one or more applications) on one DTU,

move to another DTU and then resume the user session exactly where the user left off. Similarly, if a DTU fails, a user can move from the failed DTU to another DTU without losing any work.

In one embodiment of the present invention, when a particular user session (801 or 802) within a shared client server (800) is disassociated with a DTU (811 or 812), one or more applications, within the particular user session, stop or reduce consumption of one or more resources from the client server (800). Specifically, the client server (800) determines when an application (803, 804, 805, 806, 807, or 808) within a user session (801 or 802) becomes inactive. A first signal is then sent to the application to indicate that the application should stop or reduce consumption of one or more resources (809) within the server (800). The server (800) then determines when the application should resume activity and sends a second signal to the application to resume or increase consumption of the one or more resources within the client server. This management of the one or more resources within the shared client server is effected transparently (e.g., via a filter), below the notice of (or independently from) the applications within the particular user session. That is, the applications are not modified within the present network environment in any way.

More specifically, the server (800) comprises a separate filter (810) that carries out the resource management functions of the applications (803, 804, 805, 806, 807, and/or 808) within the user session (801 or 802). This filter (810) is used to filter out one or more (or a list) of the applications that should stop or reduce consumption of the one or more resources (809). The present filter (810) is an asynchronous resource management mechanism because it does not alter the executable instructions of the applications to carry out its resource management functions and does not have to intercept the applications at the entry points. That is, the present filter (810) is separated from and operates alongside the functions of the applications and reaches into the applications to control their resource consumption. Thus, the applications are not modified in any way. In addition, the present invention can operate on any

application without relying on particular interactions with the application (e.g., with an operating system or a window system). That is, by using a separate filter to control the resource consumption of an application, the present invention can manage resource allocations for complex applications or sets of applications.

The Applicants have amended Claims 20, 25, 29, 31, 34, and 36 to better clarify certain features of the subject matter being claimed.

The Examiner rejected Claims 20-39 under 35 U.S.C. § 103(a) over Spilo in view of Susai. These rejections are respectfully traversed.

Spilo discloses "a method for reducing the memory requirements and CPU cycle consumption of an executing program . . . by intercepting the entry points of the program." See first sentence of the Abstract; see also Col. 2, lines 64-68. Thus, Spilo only teaches how to reduce resource consumption of an application (or program) by intercepting it at its entry point. Thus, Spilo relies on a synchronous interposer in the executable path of the program (or application) that is having its resource managed. That is, in Spilo, if the application does not execute through (i.e., from beginning to end) the instructions of the interposer's functions, then Spilo's resource management method will fail.

By contrast, as mentioned above, the resource managed application (or software) of the present invention are not modified (e.g., compressed) in any way. Rather, the present invention includes, for example, the use of a separate filter located on a client server to stop (or slow down) or start (or speed up) the consumption of a resource on the client server by the application. That is, the filter of the present invention is an asynchronous resource management mechanism that does not alter the executable instructions of the application (or window system or operating system) to call out its resource management function. The filter operates alongside the functions of the application and reaches in to control the application's resource consumption. Thus, the mechanism for carrying out the resource management function of the present invention is completely different from Spilo (which is synchronous).

In addition, Spilo is directed to nothing more than a traditional state machine (i.e., a Windows or a Windows 95 based personal computer) that may not even be associated with a traditional network computer architecture. The present invention, on the other hand, is directed to a **stateless** machine in a shared client computing architecture. Accordingly, Spilo fails to disclose, teach, or suggest the stateless machine (i.e., the DTU) of the present invention.

More specifically, with respect to independent Claim 31, Spilo fails to disclose or suggest a client server serving a plurality of applications to a stateless Desktop Unit (DTU) comprising:

> **a filter for managing consumption of said resource;**
>
> **wherein said filter is separated from said plurality of applications;**
>
> a first session associated with a user on a first **stateless DTU**;
>
> wherein said first session is disassociated with said first DTU, indicating that said first session is inactive;
>
> a first signal transmitted from said filter to at least one member of said plurality of applications indicating that said at least one member should stop consuming said resource;
>
> wherein said first session associated with said user becomes re-associated with **any stateless DTU**, indicating that said session has resumed activity; and
>
> a second signal transmitted from said filter to said at least one member indicating that said at least one member should resume consuming said resource (Emphasis in underline added).

Similar limitations, which are neither disclosed in nor suggested by the cited references, are present in amended independent Claims 20 and 36. Moreover, with regard to independent Claims 20 and 36, the Examiner acknowledges that Spilo fails to

disclose "when said application served from said client server should resume activity." To make up for this deficiency, the Examiner proposes the combination of Spilo with Susai. Susai discloses a method for a client to be connected to a server. Specifically, in Susai, rather then having a client be directly connected to the server, the client is first connected to an interface unit 202. The interface unit 202 then connects the client to the server. When the client is disconnected from the server, the interface unit 202 is not disconnected from the server. The interface unit 202 maintains the open connection with the server so that the client can be connected with the server more efficiently. That is, the client and server do not have to "exchange three packets of information to setup a connection." See Col. 1, lines 50-52. Thus, because Susai addresses a completely different problem of efficiently connecting a client to a server without having to exchange the same connection information ("connection loading") each time the client is connected with the server via an intermediate interface unit 202 and not reducing consumption of a resource by an application executing within the server, there is no motivation to use Susai in combination with Spilo to teach "when said application served from said client server should resume activity." Accordingly, the Applicants respectfully believe that the motivation could only come from the advantages taught and suggested in the present application; thus proper grounds for an obviousness rejection are absent with regard to the claims in the present application (i.e., hindsight reconstruction).

Furthermore, even if such a teaching or suggestion for the proposed combination were present, the cited references still fail to disclose "determining when said application served from said client server should resume activity" and from this determination "sending a second signal to said application . . . to indicate that said application should resume or increase consuming said one or more resources." See Claim 20; see also Claim 36. Regardless, Susai otherwise fails to make up for the deficiencies in Spilo cited above.

Moreover, amended Claim 20 contains the further recitations of:

> **filtering said application from said plurality of applications served from said client server via a filter located within said client server and separated from said plurality of applications;**
>
> sending a first signal to said application served from said client server to indicate that said application should stop or reduce consuming said one or more resources on said client server **via said filter**;
>
> * * *
>
> sending a second signal to said application served from said client server to indicate that said application should resume or increase consuming said one or more resources on said client server **via said filter** (Emphasis in underline added).

Amended Claim 36 is directed to a computer program product for improving access to one or more resources on a plurality of servers and comprising among other things:

> computer readable program code configured to cause a filter on at least one of said plurality of client servers to filter said application from said plurality of application;
>
> computer readable program code configured to cause at least one of said plurality of client servers via said filter to send a first signal to said application indicating that said application should stop or reduce consuming said one or more resources;
>
> computer readable program code configured to cause at least one of said plurality of client servers to determine when said application should resume activity; and
>
> computer readable program code configured to cause at least one of said plurality of client servers via said filter to send a second signal to said application indicating that said application should resume or increase consuming said one or more resources.

Accordingly, a *prima facie* case of obviousness has not been established and the rejection of independent Claims 20, 31, and 36 should be withdrawn. In addition, the rejection of Claims 21-30, 32-35, and 37-39, which depend from Claims 20, 21, or 62, should be withdrawn. In addition, with the impermissible use of hindsight, the Examiner also argues that Spilo (and Susai) can be combined with Tushie to disclose the use of the "smart card" defined in Claims 24 and 33. First, it should be noted that Tushie discloses nothing more than a method for making a personalized smart card. Thus, there is no teaching or suggestion in Tushie of using its "smart card" in combination with the Windows based personal computer resource management approach in Spilo (which should already be personalized). Regardless, the addition of Tushie otherwise still fails to make up for the deficiencies in Spilo (and Susai) cited above.

Moreover, for example, Claim 30 should be independently allowable because the cited references do not disclose or suggest a client server providing "a computational power for said stateless DTU and a state maintenance for said stateless DTU." Likewise, the cited references fail to disclose or suggest "wherein said **any** stateless DTU comprises said first stateless DTU and a second stateless DTU" as defined in Claim 32 and "wherein said first and second signals are sent by said first client server comprising said filter, and wherein said plurality of applications are served by said second client server," as defined in Claim 34.

In view of the foregoing, the Applicants respectfully submit that Claims 20-39 are in condition for allowance. Reconsideration and withdrawal of the rejections is respectfully requested, and a timely Notice of Allowability is solicited. To the extent it would be helpful to placing this application in condition for allowance, the Applicants encourage the Examiner to contact the undersigned counsel and conduct a telephonic interview.

The Commissioner is authorized to charge $770.00 for request for continued examination (RCE) pursuant to 37 CFR § 1.17(e) and any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-0639.

Respectfully submitted,

Date: October 17, 2003

Peter C. Hsueh
Attorney for Applicants
Registration No. 45,547

O'MELVENY & MYERS LLP
400 South Hope Street
Los Angeles, CA 90071-2899
Telephone: (213) 430-6000